PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-097203
(43)Date of publication of application : 08.04.1997

(51)Int.Cl. G06F 12/00
G06F 9/46
G06F 11/34
G06F 15/16

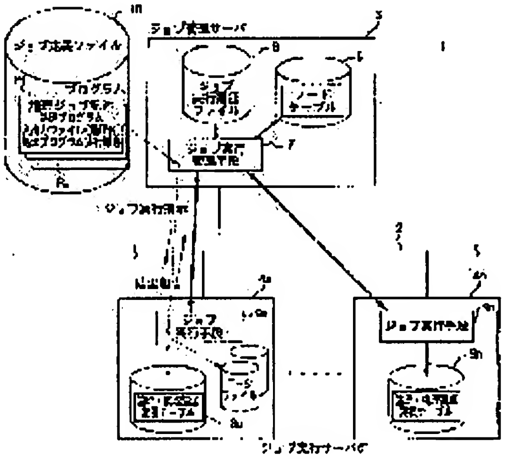
(21)Application number : 07-253076
(22)Date of filing : 29.09.1995

(71)Applicant : NRI & NCC CO LTD
(72)Inventor : YAMAMOTO SOICHI
UENO AYUMI
MUTO YOSHITO
YOSHIDA NOBUKATSU
INADA YOICHI
YAMAWAKI MASANORI
FUJISHIMA MIHO

L

(54) DISTRIBUTED PROCESSING SYSTEM

(57)Abstract:



PROBLEM TO BE SOLVED: To provide a distributed processing system with which the descriptions of programs by systematically unified logical file titles are permitted and a uniform job execution history can be outputted.

SOLUTION: This system is provided with a job managing server 3 for managing entire distributed processing and job executing servers 4a...4n for executing the jobs of programs and the programs to be distributedly processed are described by processing programs, the execution order of processing programs and logical title files. Then, the job managing server 3 is provided with a node table 5 recording the names or the like of job executing servers, a job execution history file 6 and a job execution managing means 7 for managing the execution of jobs, and the job executing servers 4a~4n are provided with logical/physical resource converting tables 8a...8n

for converting the logical title files into physical resource files in reading and job executing means 9a...9n for executing jobs.

LEGAL STATUS

[Date of request for examination]	19.03.2002
[Date of sending the examiner's decision of rejection]	
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	3795107
[Date of registration]	21.04.2006
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	

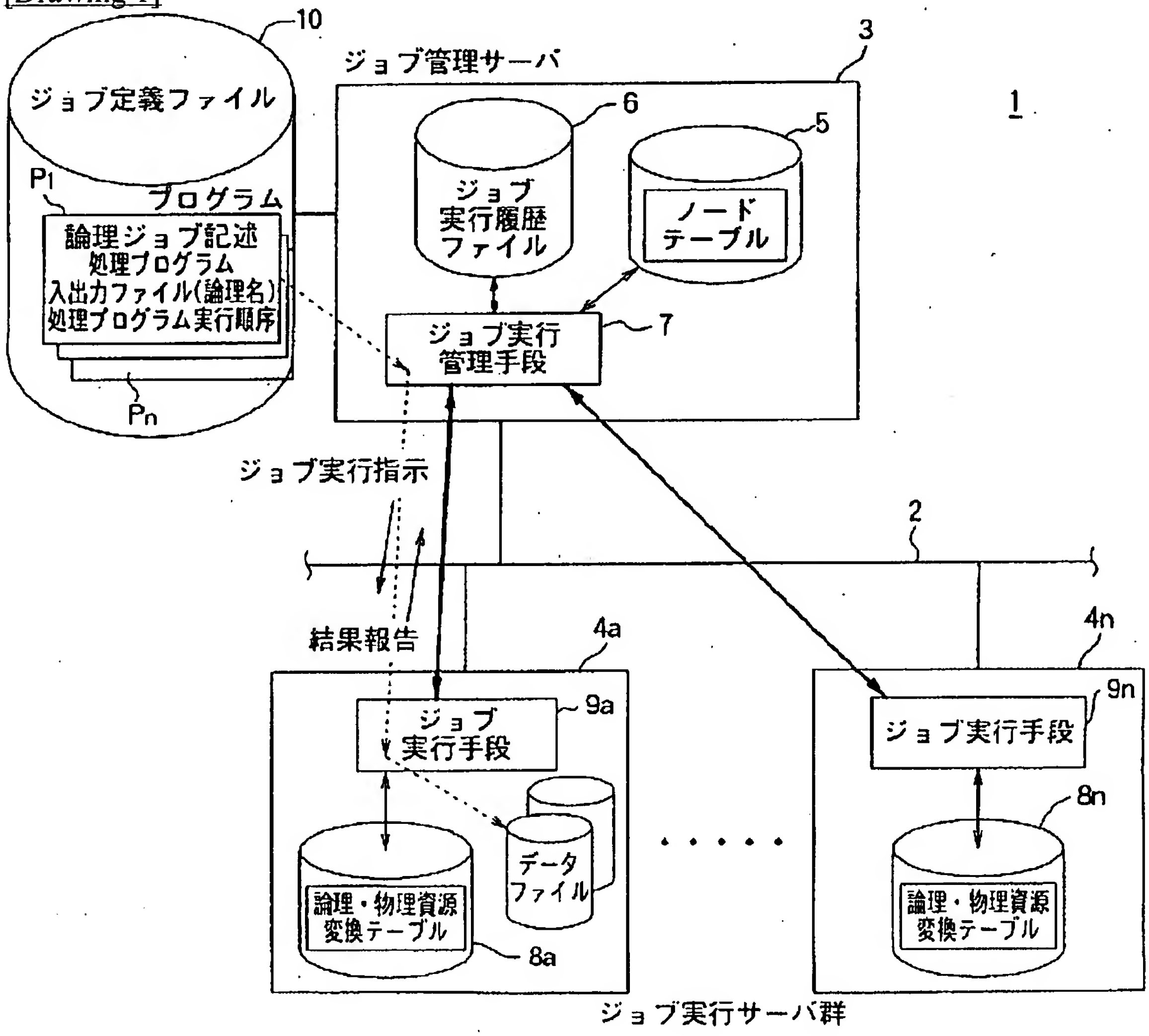
* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

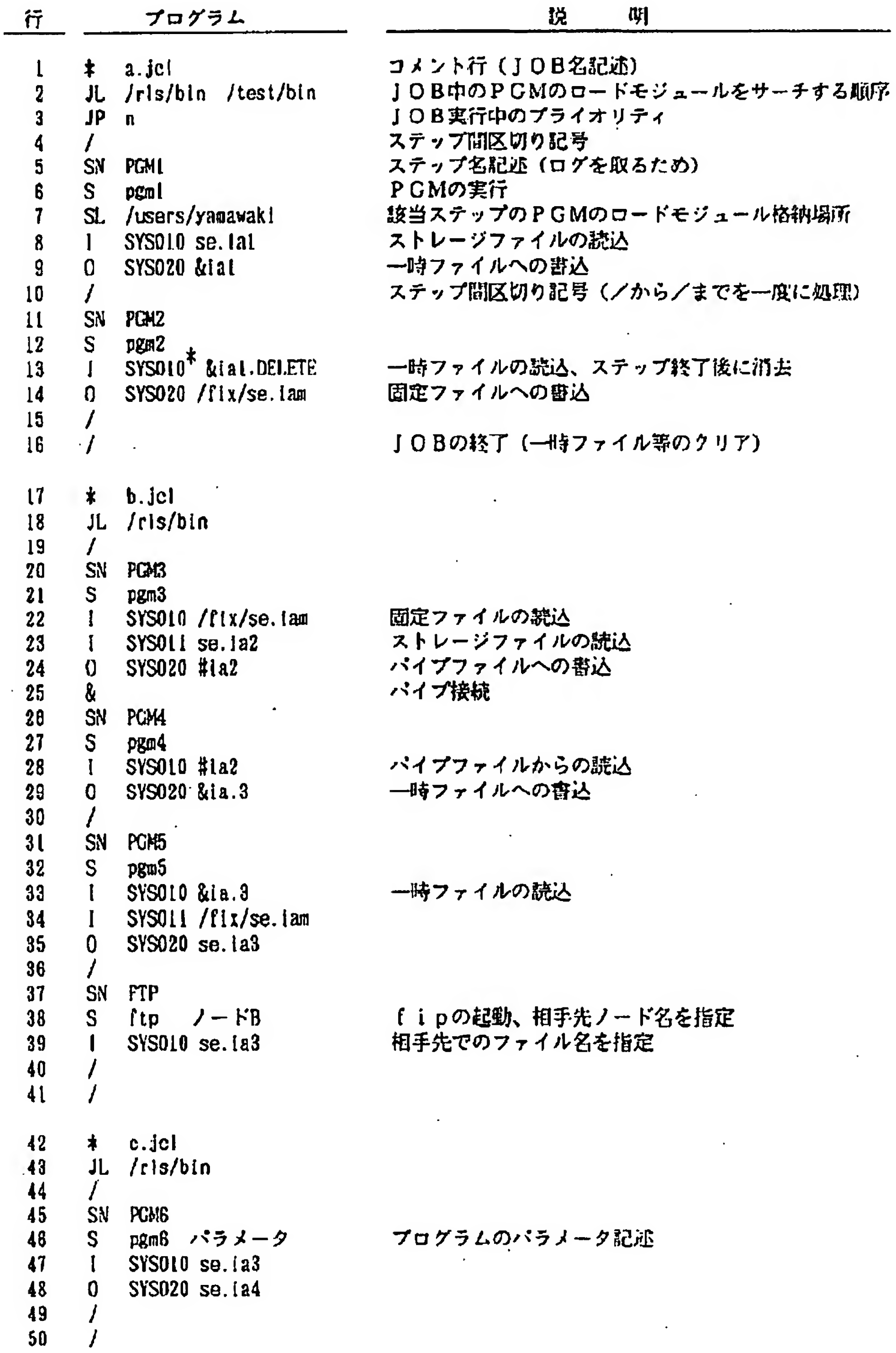
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DRAWINGS

[Drawing 1]

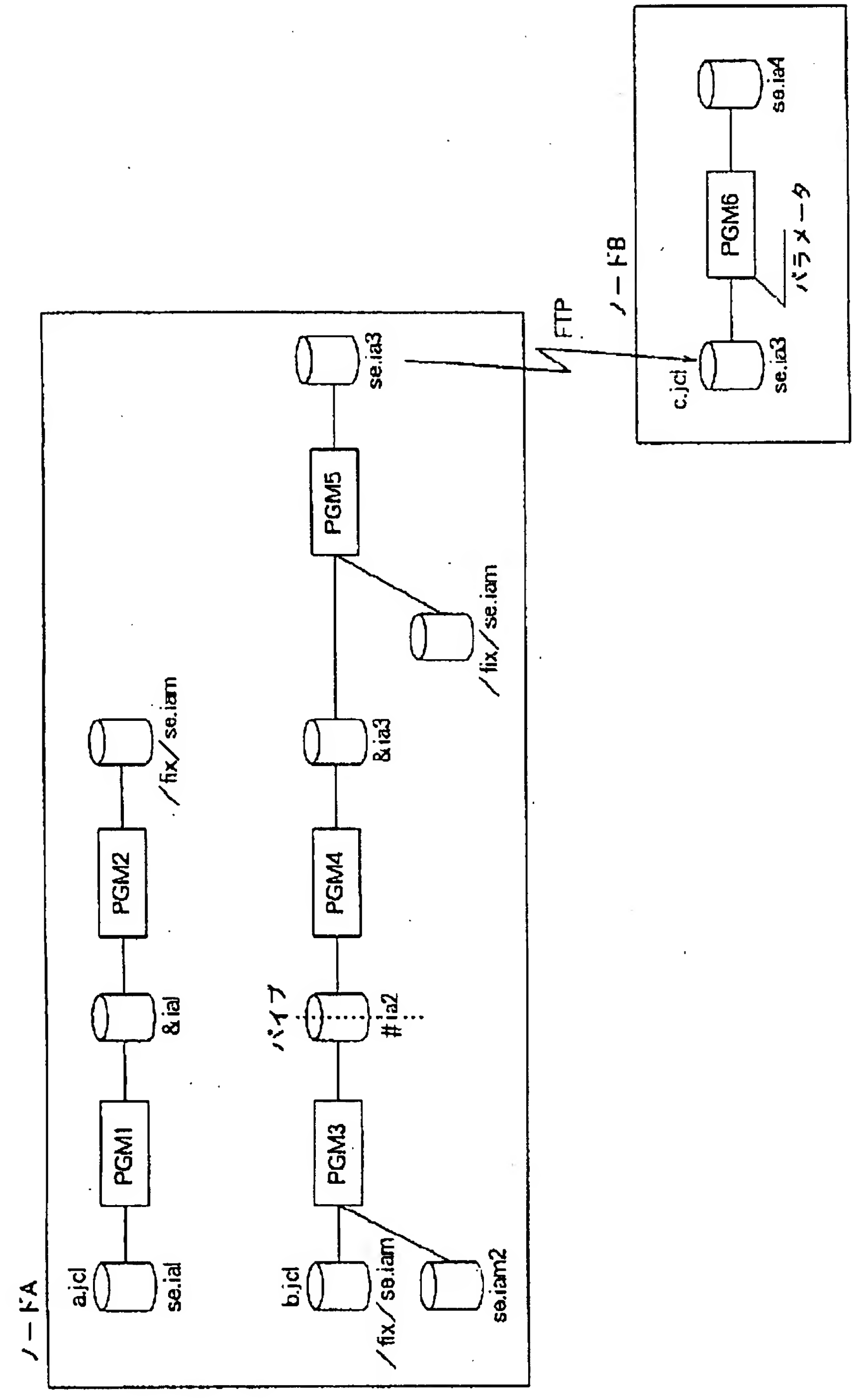


[Drawing 2]

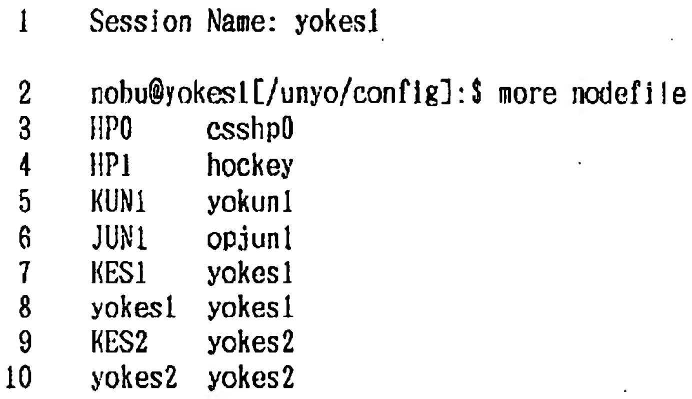


行	プログラム	説　明
1	*　a.jcl	コメント行（ＪＯＢ名記述）
2	JL　/rls/bin　/test/bin	ＪＯＢ中のＰＧＭのロードモジュールをサーチする順序
3	JP　n	ＪＯＢ実行中のプライオリティ
4	/	ステップ間区切り記号
5	SN　PGM1	ステップ名記述（ログを取るため）
6	S　pgm1	ＰＧＭの実行
7	SL　/users/yamawaki	該当ステップのＰＧＭのロードモジュール格納場所
8	I　SYS010 se.ia1	ストレージファイルの読込
9	O　SYS020 &ia1	一時ファイルへの書込
10	/	ステップ間区切り記号（／から／までを一度に処理）
11	SN　PGM2	
12	S　pgm2	
13	I　SYS010* &ia1.DELETE	一時ファイルの読込、ステップ終了後に消去
14	O　SYS020 /fix/se.iam	固定ファイルへの書込
15	/	
16	/	ＪＯＢの終了（一時ファイル等のクリア）
17	*　b.jcl	
18	JL　/rls/bin	
19	/	
20	SN　PGM3	
21	S　pgm3	
22	I　SYS010 /fix/se.iam	固定ファイルの読込
23	I　SYS011 se.ia2	ストレージファイルの読込
24	O　SYS020 #ia2	パイプファイルへの書込
25	&	パイプ接続
26	SN　PGM4	
27	S　pgm4	
28	I　SYS010 #ia2	パイプファイルからの読込
29	O　SYS020 &ia.3	一時ファイルへの書込
30	/	
31	SN　PGM5	
32	S　pgm5	
33	I　SYS010 &ia.3	一時ファイルの読込
34	I　SYS011 /fix/se.iam	
35	O　SYS020 se.ia3	
36	/	
37	SN　FTP	
38	S　ftp　ノードB	ｆｉｐの起動、相手先ノード名を指定
39	I　SYS010 se.ia3	相手先でのファイル名を指定
40	/	
41	/	
42	*　c.jcl	
43	JL　/rls/bin	
44	/	
45	SN　PGM6	
46	S　pgm6　パラメータ	プログラムのパラメータ記述
47	I　SYS010 se.ia3	
48	O　SYS020 se.ia4	
49	/	
50	/	

[Drawing 3]


ノードA
a.jcl
se.ial
PGM1
&ial
PGM2
/fix/se.iam
b.jcl
/fix/se.iam
se.iam2
PGM3
パイプ
#ia2
PGM4
&ia3
/fix/se.iam
PGM5
se.ia3
FTP
ノードB
c.jcl
se.ia3
PGM6
パラメータ
se.ia4

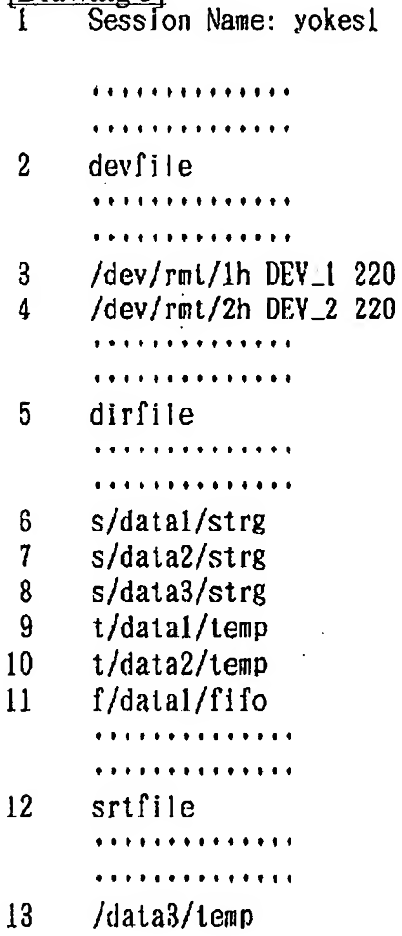
[Drawing 4]



```
1    Session Name: yokes1

2    nobu@yokes1[/unyo/config]:$ more nodefile
3    HP0      csshp0
4    HP1      hockey
5    KUN1     yokun1
6    JUN1     opjun1
7    KES1     yokes1
8    yokes1   yokes1
9    KES2     yokes2
10   yokes2   yokes2
```

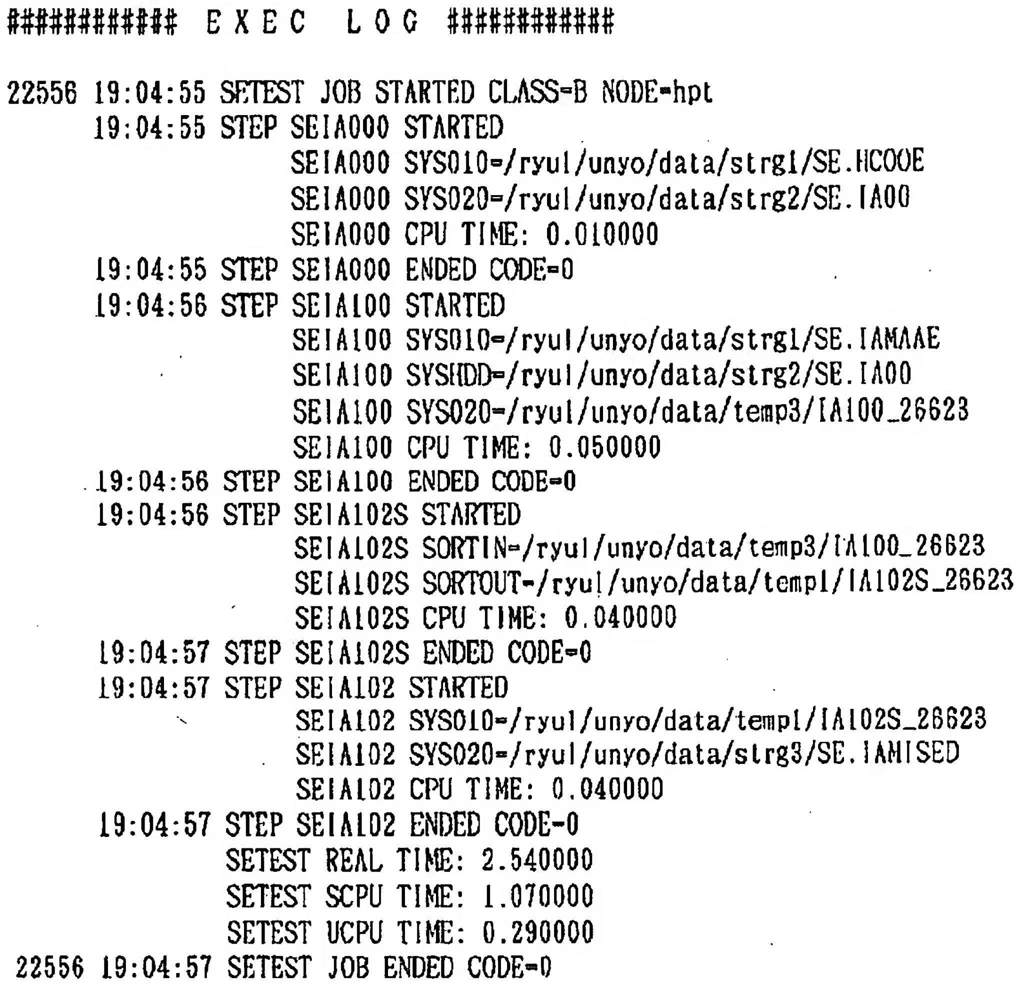
[Drawing 5]



```
1    Session Name: yokes1

     ..............
     ..............
2    devfile
     ..............
     ..............
3    /dev/rmt/1h DEV_1 220
4    /dev/rmt/2h DEV_2 220
     ..............
     ..............
5    dirfile
     ..............
     ..............
6    s/data1/strg
7    s/data2/strg
8    s/data3/strg
9    t/data1/temp
10   t/data2/temp
11   f/data1/fifo
     ..............
     ..............
12   srtfile
     ..............
     ..............
13   /data3/temp
```

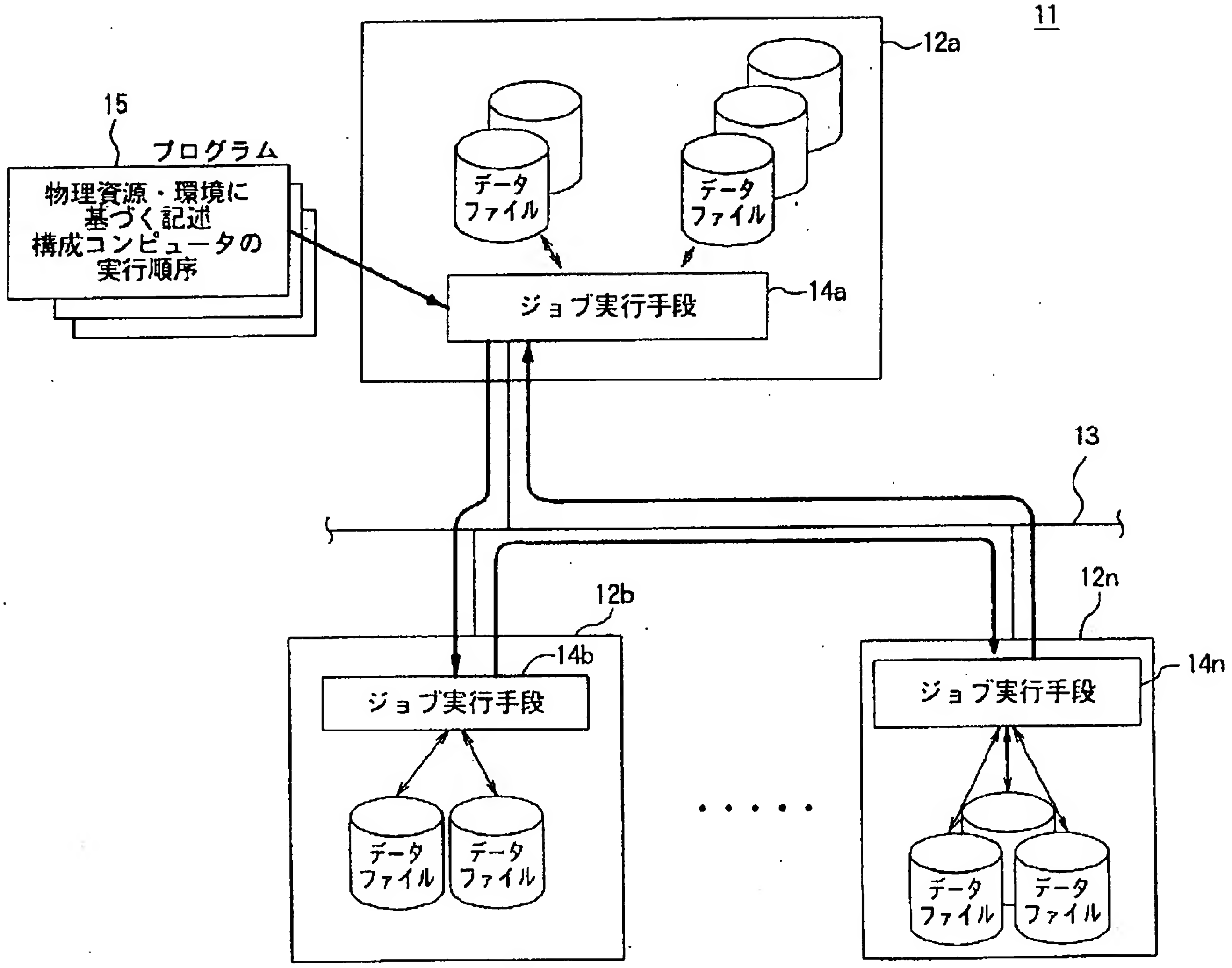
[Drawing 6]



```
########### EXEC LOG ###########

22556 19:04:55 SETEST JOB STARTED CLASS=B NODE=hpt
      19:04:55 STEP SEIA000 STARTED
                    SEIA000 SYS010=/ryul/unyo/data/strgl/SE.HCOOE
                    SEIA000 SYS020=/ryul/unyo/data/strg2/SE.IA00
                    SEIA000 CPU TIME: 0.010000
      19:04:55 STEP SEIA000 ENDED CODE=0
      19:04:56 STEP SEIA100 STARTED
                    SEIA100 SYS010=/ryul/unyo/data/strgl/SE.IAMAAE
                    SEIA100 SYSHDD=/ryul/unyo/data/strg2/SE.IA00
                    SEIA100 SYS020=/ryul/unyo/data/temp3/IA100_26623
                    SEIA100 CPU TIME: 0.050000
      19:04:56 STEP SEIA100 ENDED CODE=0
      19:04:56 STEP SEIA102S STARTED
                    SEIA102S SORTIN=/ryul/unyo/data/temp3/IA100_26623
                    SEIA102S SORTOUT=/ryul/unyo/data/templ/IA102S_26623
                    SEIA102S CPU TIME: 0.040000
      19:04:57 STEP SEIA102S ENDED CODE=0
      19:04:57 STEP SEIA102 STARTED
                    SEIA102 SYS010=/ryul/unyo/data/templ/IA102S_26623
                    SEIA102 SYS020=/ryul/unyo/data/strg3/SE.IAMISED
                    SEIA102 CPU TIME: 0.040000
      19:04:57 STEP SEIA102 ENDED CODE=0
               SETEST REAL TIME: 2.540000
               SETEST SCPU TIME: 1.070000
               SETEST UCPU TIME: 0.290000
22556 19:04:57 SETEST JOB ENDED CODE=0
```

[Drawing 7]



[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Field of the Invention] This invention allows description by the logical file name unified systematically, without being conscious of the physical storage name of the computer actually connected etc. about the program which should relate to the distributed processing system which connects two or more computers and carries out distributed processing of the program, especially should be processed, and relates to the distributed processing system in which an output of job activation hysteresis is possible about the processing result of a program.
[0002]
[Description of the Prior Art] Conventionally, processing of a big program was systematically processed by large-sized computer called a mainframe. However, recently, the demand which is going to perform processing equivalent to a large-sized computer was increasing from the demand of the flexibility of a system, economical efficiency, the dependability of a system, etc. by connecting a minicomputer, a workstation, a personal computer, etc. by the communication line, and carrying out distributed processing of the program.
[0003] Recently in response to this demand, various distributed operating systems (it is called distributed process input output equipment OS) are proposed. According to these distributed process input output equipment OS, a program can be described by the unified syntax and the file in other computers can be accessed now. This distributed process input output equipment OS enabled the large-scale program technically to carry out distributed processing by two or more computers. As such distributed process input output equipment OS, there is UNIX, for example.
[0004] Drawing 7 shows the system configuration of the hardware in the case of carrying out distributed processing of the program by conventional distributed process input output equipment OS.
[0005] As shown in drawing 7 , the conventional distributed processing system 11 consists of what connected two or more computers 12a, --, 12n by the communication line 13.
[0006] Although each computers 12a, --, 12n have the configuration of a physical resource different, respectively, for example, the store which stores a data file, they are juxtaposition-like mutually in common at the point of processing a program with the same job activation means 14a, --, 14n.
[0007] The program 15 processed with the distributed processing system 11 of the above-mentioned configuration must be described based on a physical resource and an environment. That is, the program 15 had to specify the job (a series of processings in a program), and the Computers [12a --, 12n] execution sequence, had to specify each Computers [12a --, 12n] concrete store name and a directory name, and must describe the I / O file.
[0008] In the example of drawing 7 , as shown in an arrow head, a program 15 is first processed by computer 12a, next, the processing result is transmitted to computer 12b, and is processed, further, it is processed by computer 12n and a final processing result is returned to computer 12a.
[0009]
[Problem(s) to be Solved by the Invention] However, it was difficult for the activity for description of a

program to deal with an escape and modification of a system configuration flexibly complicated in the above-mentioned conventional distributed processing system.

[0010] That is, although description of the program by the above-mentioned distributed process input output equipment OS could be described according to the unified syntax, the programmer had to describe clearly the whereabouts of data, the processing at the time of error generating, etc. with those physical equipments, bearing in mind the physical device name of each computer which constitutes a system. Thus, it had become a big burden for a programmer to describe the physical whereabouts one by one about the I / O file in a program. Especially, in the large-scale system, a programmer's burden mentioned above increased remarkably.

[0011] Moreover, the program which carried out in this way and was created is effective only in the system configuration of a proper, and was not able to be used for the distributed processing system of a configuration of differing. Therefore, in such a distributed processing system, when adding and changing the computer which constitutes a system, or when the equipment configuration of each computer was changed, the contents of the program had to be improved again. Now, the flexibility which is the advantage of a distributed processing system was not fully able to be harnessed.

[0012] Furthermore, with the conventional distributed processing system, since the processing result was exchanged between the computers of a system, when trouble was in processing of computer 12b and a processing result was not transmitted to computer 12n, for example, it was not able to specify whether trouble occurred as which part of a program, so that clearly from the example of drawing 7 .

[0013] Moreover, even when all the jobs were processed harmoniously even if, the data of how many which computers worked were not able to be obtained.

[0014] When two or more programs were especially processed in concurrency, the load of each computer was not able to be taken into consideration.

[0015] Then, the technical problem which this invention tends to solve is in distributed system to offer the distributed processing system which allows description of the program by the logical file name unified systematically, and can obtain the processing result more than a large-sized computer and an EQC.

[0016]

[Means for Solving the Problem] In order to solve the above-mentioned technical problem, the distributed processing system concerning this application claim 1 In the distributed processing system which connects two or more computers and carries out distributed processing of the program One of the connected computers is made into the job management server which manages the whole distributed processing of a program. The processing program which considers as the job activation server which performs the job which consists other computers of a series of processings in said program, respectively, and performs respectively fixed processing for said program which should be carried out distributed processing, The node table which considered as the program by the logic job description which consists of an execution sequence of said processing program, and a I / O file by the logical name, and recorded the identifier and machine address of a job activation server on said job management server, The job activation history file which records the job activation result by each job activation server, A job execution control means to carry out the distribution directions of the activation of a job at each job activation server, and to record those processing results on said job activation history file, It is characterized by having the logic and the physical-resource translation table which reads and changes the I / O file by said logical name to the file on the physical resource of each job activation server at a preparation and said job activation server, and a job activation means to perform a job.

[0017] Moreover, it is characterized by the DPS concerning this application claim 2 outputting the job list with which said job management server includes the activation start time of each job and end time, a I / O file, and the activation result of a job with reference to said job activation history file in the DPS of above-mentioned claim 1.

[0018] In the above-mentioned DPS, the program in which the job execution control means of a job management server carried out logic job description is read, and activation of the job which constitutes a program is distributed and directed at each job activation server.

[0019] The job activation server to which activation of a job was directed from the above-mentioned job management server actually performs a job with a job activation means, after reading and changing the logical name file described by the program with logic and a physical-resource translation table to the file on the physical device name which the job activation server has.
[0020] The job activation server which ended activation of a job reports the activation result of a job to a job management server. The job management server which received the report of this job activation result registers the activation result of each job into a job activation history file.
[0021] Furthermore, a job management server can output a job list including the activation hysteresis of a job, i.e., the activation start time of each job, end time, a I / O file, the activation result of each job, etc. with reference to a job activation history file by demand.
[0022]
[Embodiment of the Invention] The distributed processing system by 1 operation gestalt of this invention is explained using the drawing of attachment in an application below.
[0023] Drawing 1 shows the system configuration and the flow of processing of this distributed processing system. It is job activation server 4a which performs the job which one of the computers which this distributed processing system 1 consisted of that to which two or more computers were connected by the communication line 2, and were connected is made into the job management server 3 which manages the whole distributed processing of a program as shown in drawing 1 , and it becomes from a series of processings in a program about other computers, respectively. -- It may be 4n.
[0024] The job management server 3 is job activation server 4a. -- To each job activation server, distribution directions were carried out to the 4n identifier, the node table 5 which recorded the machine address, and the job activation history file 6 which records the job activation result by each job activation server, activation of a job was managed with it, and it has a job execution control means 7 to record those processing results on the job activation history file 6.
[0025] Job activation server 4a -- 4n is the logic and physical-resource translation table 8a which reads and changes the I / O file by the logical name to the file on the physical resource of each job activation server. -- It is job activation means 9a which actually performs a job and reports that an activation result is 8n to the job management server 3. -- It has 9n.
[0026] In addition, in this DPS 1, it has the job definition file 10 which supplies the program which should be carried out distributed processing to the job management server 3. This job definition file 10 is the program (this description approach is called logic job description on these specifications) P1 described by the processing program name which is memorized by the predetermined storage of the interior of a system, or the exterior, and performs fixed processing, the execution sequence of these processing programs, and the I / O file by the logical name. -- Pn A large number are stored.
[0027] program P1 --Pn by which logic job description was carried out [above-mentioned] in this DPS 1 It is sent to the job execution control means 7 of the job management server 3 from the job definition file 10. The job execution control means 7 program P1 --Pn description -- following -- program P1 --Pn the job (a series of processings in a program) to constitute -- each job activation server 4 -- it distributes to a--4n, and the activation is directed.
[0028] Job activation server 4a -- 4n job activation means 9a -- 9n is logic and physical-resource translation table 8a. -- With reference to 8n, the logical name file in a program is read and changed to the file on each physical resource, and a job is performed.
[0029] if processing of a job is completed -- each job activation server 4 -- the a--4n job activation means 9 -- a--9n (on these specifications, a job activation result points out time amount, a I / O file, etc. which were spent on the signal of whether the job was processed convenient and activation of a job) of activation results of a job is reported to the job execution control means 7 of the job management server 3. In response, the job execution control means 7 registers the activation result of each job into the job activation history file 6. the data (this is called "processed data" in distinction from a "job activation result" on these specifications) obtained by activation of a job -- each job activation server 4 -- from a--4n, it is sent to the job management server 3, and is outputted through the output means which is not illustrated.

[0030] next, program P1 which should be carried out distributed processing with this distributed processing system 1 or [description of --Pn being simplified how, and having an escape and flexibility, and realizing processing equivalent to a large-sized mainframe] -- program P1 --Pn etc. -- an example is shown and it explains below.
[0031] Drawing 2 indicates the explanation about each of that sentence to be an example of the program which is used for this DPS 1 and by which logic job description was carried out. Drawing 3 is the flow Fig. having shown the processing flow of the program of drawing 2 notionally.
[0032] As shown in drawing 2 , this program by which logic description was carried out has described three jobs, a.jcl (- of 1st line the 16th line), b.jcl (- of 17th line the 41st line), and c.jcl (- of 42nd line the 50th line). Each job consists of combination of the batch of the "step" divided by "/."
[0033] If it observes about the job of "a. jcl" now, the job name (the 1st line), the search order (the 2nd line) of the storing location of the load module of PGM.(the abbreviation for a program), and the execution priority (the 3rd line) of a job will be described first.
[0034] Next, processing of a step (- of 4th line the 10th line) is described. Processing of this step describes [next] a step name first (the 5th line). Activation of the processing program "pgm1" which performs fixed processing is directed (the 6th line). Next, after directing the storing location of the program of the low order of the 2nd line search order, the storage file "SYS10 se.ial" to input and temporary file "SYS20 &ial" outputted (the 8th line) of (the 7th line) are directed. Here, "SYS10" and "SYS20" are general agreements which show an input and an output file, and "se.ial" and "&ial" are the proper names of each file.
[0035] Input a storage file "se.ial", and with a processing program "pgm1", the semantics of the above-mentioned step should process a storage file "se.ial", and should output a processing result to temporary file "&ial" (refer to drawing 3).
[0036] the result of having directed activation of a program "pgm2" (the 12th line), having inputted temporary file "&ial" outputted previously at the following step (- of 10th line the 15th line) (the 13th line), and having processed temporary file "&ial" by the program "pgm2" -- fixed file "/fix/se.iam . common to a system -- " -- writing in (the 14th line) -- things are directed. Processing of a job "a. jcl" is ended by the two above-mentioned steps.
[0037] A fixed file "/fix/se.iam" and a storage file "se.ia2" are inputted. the same -- b. "jcl" -- drawing 3 -- an understanding -- so that it may be shown easily It processes by the program "pgm3", and for every data, a processing result is delivered to a program "pgm4" on memory (this is called pipe function), and is processed by the program "pgm4", and a processing result is written in temporary file "&ia3." Next, temporary file "&ia3" and a fixed file "/fix/se.iam" are read, it processes by the program "pgm5", and a processing result is written in a storage file "se.ia3." Furthermore, this storage file "se.ia3" is transmitted to the storage file "se.ia3" of Node B. Here, Node B shows the specific computer connected to the distributed processing system, and the name of a specific job activation server is filled in in practice.
[0038] Processing of a job "b. jcl" is ended above. In addition, although b. the above-mentioned job "a. jcl" and "jcl" are unconditionally processed by Node A since they are written to the beginning of a program, they are performed by the specified node B from the following job "c. jcl."
[0039] A job "c. jcl" inputs a storage file "se.ia3", processes this file by the program "pgm6", and is directing to write a processing result in a storage file "se.ia4."
[0040] The description on description of the above-mentioned program is explained below. the program processed with this distributed processing system 1 -- "se.ia1" and -- like "se.ia4" (storage file), "&ia1", --, "&ia3" (temporary file), and "#ia2" (pipe file), the logical file name defined systematically beforehand is used.
[0041] This makes description of the conventional program, and a big difference. That is, in consideration of the configuration of the physical resource of the above-mentioned I / O file and the computer (here, they are Node A and Node B) by which, as for "temporary file", a "pipe file", etc., a programmer especially actually performs a job, the specific directory (storing location of file) name of the specific store name of each computer had to be specified concretely conventionally.
[0042] Therefore, the store name and directory name which are different for every job activation server

for every file had to be investigated, and this had become a big burden on a programming. The same was said of the node name (the identifier of a job activation server, machine address).
[0043] On the other hand, the program processed with this DPS 1 is first divided into a "storage file", "temporary file", and a "pipe file" about a file, and if the system of the file name which attached a predetermined name and a predetermined serial number is defined for every file of various kinds, a programmer can create a program, without being conscious of the physical environment of a job activation server.
[0044] the node name which described the job execution control means 7 to the program with reference to the node table 5 when the above-mentioned program was read into the job execution control means 7 of the job management server 3 -- following -- each job -- each job activation server 4 -- it distributes to a--4n, and activation is directed.
[0045] Here, an example of the node table 5 is shown in drawing 4 . it is shown in drawing 4 -- as -- the node table 5 -- each job activation server 4 -- an a--4n machine address and a device name are listed. On this node table 5, the node name described by the program can be searched and each job activation server can be specified.
[0046] Job activation server 4a which received the job run command from the job execution control means 7 -- 4n is logic and physical-resource translation table 8a. -- With reference to 8n, a logical name file is read and changed to the file on the physical equipment of each job activation server, and a job is performed.
[0047] They are logic and physical-resource translation table 8a here. -- A 8n example is shown in drawing 5 . - of 6th line the 8th line of drawing 5 shows the physical device name or physical directory name which stores a storage file. Similarly, eye said 11 lines [the physical unit name and directory name in which the 9th line of drawing 5 and the 10th line store temporary file, and] shows the physical unit name and the directory name which stores a pipe file, respectively.
[0048] The above-mentioned logic and physical-resource translation table 8a -- By 8n, it is job activation server 4a. -- A file can be treated on each physical unit and 4n of jobs can be performed convenient.
[0049] Thus, job activation server 4a which performed the job -- Reporting 4n of activation results of a job to the job execution control means 7, the job execution control means 7 registers the activation result of these jobs into the job activation history file 6.
[0050] By referring to this job activation history file 6, DPS 1 by this invention can output the same job list as large-sized computers, such as a main frame.
[0051] A job list can be greatly useful to a program verification including the activation start time of each job and end time, a I / O file, and the activation result of a job.
[0052] Drawing 6 shows an example of a job list. As shown in drawing 6 , information, such as a time of the activation start time of each job (this list SEIA000, SEIA100, etc.), activation end time, a I / O file, and CPU, can be acquired by referring to a job list. When processing is interrupted by the defect of a program etc., by activation of which job, it should specify whether processing was interrupted and it should be used for discovery of the error of a program.
[0053] According to the distributed processing system of this invention, the program shows only the thought (flow of the abstracted processing) of processing, and is not influenced by the physical environment of the computer connected to the system. That is, it can respond by modification of some of node tables, and logic and physical-resource translation tables, without rewriting a program, when adding a job activation server to the DPS or, changing the configuration of the store of a predetermined job activation server for example.
[0054] Moreover, although the above-mentioned explanation explained the case where one program was processed, the distributed processing system of this invention can process two or more programs in concurrency not only like processing of one program but like a large-sized computer.
[0055] in this case, the job execution control means 7 -- program P1 --Pn or [that it is the job which follows and belongs to which program] -- not related -- the execution sequence of a job -- giving -- one after another -- each job activation server 4 -- activation of a job is directed to a--4n.

[0056] each job activation server 4 -- in a--4n, an unsettled job will be in the condition of waiting and serial processing of the job will be carried out according to an execution sequence. The processed job is reported to the job management server 3, and the job management server 3 outputs a processing result and processed data for every program.

[0057]

[Effect of the Invention] According to the distributed processing system of this invention, without being conscious of the physical resource of the connected computer, by the logical file name, a program can be described and the burden on a programming can mitigate sharply so that clearly from the above-mentioned explanation.

[0058] Moreover, the structure of a system can be extended and changed by making local correction of a node table, logic, a physical-resource translation table, etc.

[0059] By the above, according to this invention, creation of a program can respond to an escape and modification of a system flexibly easily, and can offer further the distributed processing system which realizes the same processing as a large-sized computer under the environment which connected the small computer.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] In the distributed processing system which connects two or more computers and carries out distributed processing of the program One of the connected computers is made into the job management server which manages the whole distributed processing of a program. The processing program which considers as the job activation server which performs the job which consists other computers of a series of processings in said program, respectively, and performs respectively fixed processing for said program which should be carried out distributed processing, The node table which considered as the program by the logic job description which consists of an execution sequence of said processing program, and a I / O file by the logical name, and recorded the identifier and machine address of a job activation server on said job management server, The job activation history file which records the job activation result by each job activation server, A job execution control means to carry out the distribution directions of the activation of a job at each job activation server, and to record those processing results on said job activation history file, The distributed processing system characterized by having the logic and the physical-resource translation table which reads and changes the I / O file by said logical name to the file on the physical resource of each job activation server at a preparation and said job activation server, and a job activation means to perform a job.
[Claim 2] Said job management server is a distributed processing system according to claim 1 characterized by outputting a job list including the activation start time of each job and end time, a I / O file, and the activation result of a job with reference to said job activation history file.

[Translation done.]